1

AF INSTITUTE OF TECHNOLOGY

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSO/ENS/91D-01

(1)

A DYNAMIC COMPUTER GRAPHICS MODEL OF SATELLITE ORBITS FOR USE IN INSTRUCTION AND ANALYSIS

THESIS

John P. Anton, Captain, USAF

AFIT/GSO/ENS/91D-01

THESIS APPROVAL

STUDENT: JOHN P. ANTON　　　　　　CLASS: GSO-91D

THESIS TITLE: A DYNAMIC COMPUTER GRAPHICS MODEL
OF SATELLITE ORBITS
FOR USE IN INSTRUCTION AND ANALYSIS

DEFENSE DATE: 25 NOVEMBER 1991

GRADE:

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Advisor	Thomas S. Kelso/ENS	Thomas Sean Kelso
Reader	Bruce W. Morlan/ENC	B L Ml

AFIT/GSO/ENS/91D-01

A DYNAMIC COMPUTER GRAPHICS MODEL OF SATELLITE ORBITS FOR USE IN INSTRUCTION AND ANALYSIS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Space Operations

John P. Anton, B.S.

Captain, USAF

December 1991

Approved for public release; distribution unlimited

Preface

The purpose of this study was to develop a dynamic computer graphics model of satellite orbits, suitable for use in instruction and analysis. The immediate need for this model was for an instructional aid in various of the space operations classes taught at the Air Force Institute of Technology. I also hoped to develop a model which would be of use for research, planning, and analysis involving satellite orbits and constellations.

The result of this research was a working, documented model, SATMAP3, containing the most important features for such a tool. Work should continue, however, to develop tools capable of further processing the output of SATMAP3 into forms more useful to specific types of analysis.

I would like to take this opportunity to thank my faculty advisor, Maj T. S. Kelso, for his help in defining the problem and writing this thesis. Further, critical parts of SATMAP3 were borrowed from an earlier program written by Maj Kelso, SAT-MAP2, without which SATMAP3 could never have been developed. I also wish to thank Maj Bruce Morlan, my thesis reader, for his help with the map file used, without which the program would be far slower. A final word of thanks is due my wife Ye, for her patience throughout this project.

John P. Anton

Table of Contents

Preface ii

Abstract iv

I. Introduction 1

  Background. 1
  Specific Problem. 3
  Sub-Objectives. 3

II. Literature Review 5

  The Need. 5
  The User Interface. 6
  Graphics and Orbital Algorithms. 9
  Summary. 11

III. Methodology 12

  Capability Selection. 12
  User Interface. 18
  Description of Algorithms. 21
  Interaction of Algorithms. 27
  Summary. 34

IV. Validation 35

  Algorithm Validation. 35
  User Interface Validation. 40
  User and Compatibility Testing. 42
  Summary. 43

V. Results 44

VI. Recommendations 47

Appendix A: User's Guide 50

  Installation. 52
  Execution. 52

Appendix B: Suggestions on Use 68

  Use in Instruction. 68
  Use in Analysis. 72

Bibliography 77

Vita 79

AFIT/GSO/ENS/91D-01

Abstract

A computer model of satellite orbits was developed to illustrate difficult orbital concepts. It was also designed to analyze the positions of a satellite constellation with respect to locations on the earth as an aid to analysts. A literature review revealed that existing methods rely on drawings, expensive computers, or are satellite system specific, but that it would be possible to develop such a tool for an IBM-compatible personal computer using the Pascal language. Model capabilities selected included 1) a menu driven user interface; 2) a screen display showing satellite locations, ground track, and an earth map in one of three reference frames; and 3) three file output formats allowing satellite position information to be dumped to ASCII files for further analysis. Model validation was performed to ensure that the satellites are accurately propagated in their orbits from the NORAD two-line element sets used as an input to the model. The model should be used in instruction to bring the dynamic nature of satellite orbits to life and add a memorable perspective on the nature of various constellations. In analysis, it should be used where convenience, availability, and ease of use are vital.

A DYNAMIC COMPUTER GRAPHICS MODEL OF SATELLITE ORBITS FOR USE IN INSTRUCTION AND ANALYSIS

I. Introduction

Background.

The Need. Instructors and analysts need a low-cost computer model to graphically display dynamic satellite orbits. Instructors at the Air Force Institute of Technology (AFIT) and Undergraduate Space Training (UST) have expressed a need for such a model to demonstrate difficult orbital concepts (6). This model would illustrate the meaning of various orbital parameters and orbit types (for example, inclination or a Molniya orbit), display typical satellite constellations, and show the effects of various perturbations on a satellite's orbit. An animated picture of various satellite orbits would greatly aid students' understanding of such concepts and be a valuable supplement to descriptions and drawings. Similarly, such a tool could be used by analysts, planners, and researchers to analyze existing or proposed satellite constellations. It could determine where satellites were (or will be) at a given time, and what sort of earth coverage a constellation provides (or will provide). It could be used to find how many Navstar Global Positioning System (GPS) satellites were in view to support an attack on

Baghdad on January 17th or if a satellite was in view of a ground station when it made its closest approach to another satellite. A computer model for a common personal computer could be readily available and flexible enough to be tailored to any of these uses (6).

Existing Methods. Existing methods rely on words and drawings, expensive computers, or are specific to a particular satellite system (6). AFIT and UST instructors currently use written descriptions and drawings when teaching orbital concepts. Such methods are two dimensional and static. This makes it difficult to portray the dynamic nature of satellite orbits and the effects which determine them. Analysts rely on these same methods. Drawings have the same disadvantages when used in instruction, and in addition are not flexible enough to adapt to new situations. Models using expensive computers are not always affordable, available, or convenient. However, IBM-compatible personal computers are widely available throughout the Air Force, so running a program on one would cost little. It would typically cost the price of a disk, the time to learn to use the program, and maybe the cost of a math coprocessor chip. Existing programs for specific satellite systems are not able to cope with the variety of situations or proposed scenarios often faced by analysts (10). A model would be most useful if it contained a varied database of orbits to draw upon and allowed the user to create his own specialized orbits and constellations. A dynamic computer graphics model could greatly improve the effectiveness of instructors and analysts in portraying the complex and interconnected effects which together determine a satellite's orbit.

Specific Problem.

It was the purpose of this research to develop a dynamic computer graphics model of satellite orbits for use in instruction and analysis. This model, called SATMAP3, started with routines developed by Dr. T.S. Kelso in 1989 for a program called SAT-MAP2, to which were added a user interface and other capabilities to make it suitable for use by instructors and analysts (7).

Sub-Objectives.

Research into this problem proceeded in the following six areas: 1) literature review; 2) capability selection; 3) user interface; 4) propagation and transformation algorithms; 5) projection and scaling algorithms; and 6) complete user package development. The first sub-objective was to review the literature for information relating to this project. Capabilities were then selected for inclusion in the computer model. The next sub-objective was to design an easy-to-use user interface. It was necessary to write the Pascal code, integrate it into the existing routines, and test the program to make sure it worked smoothly. At this point algorithms for orbit propagation and transformations were developed or adapted. The Pascal code to do this was written, integrated into the existing routines, and validated. Similarly, projection and scaling algorithms were also written, integrated, and validated. From this point, a complete user package was developed to include the program, database, and instructions, and the development effort documented. Since this project was extensive, it was critical to use proven methods and algorithms wherever possible, and hence a review of the literature was the first step.

II. Literature Review

The paragraphs which follow review the literature for information relating to this project. They cover the following areas: 1) the need for such a model; 2) the design of user interfaces; and 3) graphics display and orbital mechanics algorithms. In addition to reviewing journals and texts, they include a review of similar computer programs and their code for information to use.

The Need.

Earlier sections described the need expressed by Kelso for a tool allowing the real-time manipulation of orbits in three dimensions to be used in instruction and analysis (6). Yorchak also felt such a tool would be valuable:

> Since orbital analysts (individuals that monitor satellites in space) must understand very complicated three-dimensional concepts and interrelationships, the use of computer graphics could significantly enhance their training by depicting satellite orbits in a '3-D' fashion. (16:423)

Yorchak described the results of Rigney and Lutz, who found that the use of interactive graphics improved student's test scores (16:423). Yip found computer graphics useful to display and interpret other three-dimensional measurements (15:3919,3922). Computer graphics definitely have potential as a way to display the output required by such a tool.

In the past, the complex calculations required to determine satellite orbits required mainframe computers or supercomputers (12:198). Recent advances in personal computer capabilities allow such calculations to be performed on personal computers (10; 5). The use of personal computers offers advantages since they are cheap and widely available (3:17). The literature shows that the need can and should be met using a graphical approach on a personal computer.

The User Interface.

The design of a user interface for a computer graphics model of satellite orbits required research into three areas: 1) organization of the interface; 2) information to be displayed; and 3) portability of the program.

Organization of the Interface. Advances in computer technology allowed graphics interfaces to replace text interfaces with interfaces that are "friendlier and easy to use" (3:19). "Screen displays allow us to formalize and structure the input and output of information" (17:53,54). Information can be input, checked, and responded to on a character-by-character, field-by-field, or screen-by-screen basis (17:53,54). Ziegler described the three basic ways inputs can be made through such an interface. Codes can be used to signal movement between screens. Direct commands can be used. Menus can be used (17:57,58). Menus tend to be easier to use for those unfamiliar with a program and direct commands easier for those who use a program regularly (11:225). Of course, menus can be combined with codes or commands.

Careful structuring of the menu is important to reduce search times, the number of mistakes, and the time to learn how to use the program (17:53). The user must be able to determine from the menu items not only what functions or commands are available, but the relationship between these functions and between the functions and the task as a whole (17:52). In particular, "task related information ... should be arranged so as to correspond to the usual sequence of processing operations" (17:59). Ziegler also suggested making a menu easier to use by reducing the volume of information on the screen and by organizing with the principles of similarity, proximity, and symmetry (17:54,55). He suggested limiting the width of a menu to 5-7 items and the depth of a menu to 2 levels, if possible, to make it easier to find desired items and harder to get lost (17:57,58). Ledgard provided a useful summary of issues to consider when designing a menu: 1) be complete; 2) use consistent phrasing; 3) select words carefully; 4) group appropriately; 5) choose what to include; 6) provide for movement between menus; 7) show how the menus relate; and 8) provide needed data (9:77).

Many existing orbit programs have less than optimum user interfaces. Eagle limited his program to text. He used simple input statements with prompts, no defaults, no error trapping, and output data in a table form (2:122-123). The existing program to be used as a base for this project has an extensive graphics output interface, but is limited to text input statements with unclear prompts (7). The PC SOAP program provides a single selection menu with selections by single key commands, but the user must toggle between the orbit display and the menu as they cannot be displayed together (10).

Information to be Displayed. A screen must contain a variety of types of information, "not just on the task in hand, but also about how to control the dialogue and to handle screen display" (17:59). Four types of information will typically be displayed: 1) task information; 2) the available functions; 3) hardware and software status; and 4) error, help, and instruction messages (17:56). Ziegler stressed the need to explain the format of the input required. A menu selection should list options available, pre-insert likely defaults, and show the functions of keys (17:62).

Portability of the Program. The variety of computers, screens, and printers available makes it difficult to develop a program that will work on all systems. Turbo Pascal is a language widely used on IBM-compatible personal computers, but a program written in this language must still deal with differences in hardware. The support of output devices (printers and screens) is one of the most difficult problems (3:20). While there are only a few types of display screens (monochrome, CGA, EGA, VGA, and so on), the number of different printer models is almost endless. Any program with a screen print capability requires print routines specific to each combination of display and printer since each printer has its own printer driver (1:12).

Graphics and Orbital Algorithms.

The development of a user interface and the addition of capabilities to the existing computer program will require the use of graphics and orbital mechanics algorithms. These algorithms will either have to be found in the literature or developed from first principles. The following paragraphs discuss some of the algorithms and issues of interest. Further algorithms may be needed, and hence further research may be required, once the specific capabilities to be added have been selected.

Graphics Algorithms. The algorithms to display three-dimensional data on a two-dimensional screen have long existed. Geometric transformations of translation, scaling, and rotation can be accomplished by matrix multiplication of the current coordinates of a point by a transformation matrix (11:115-117). Such transformations would be required to zoom in or out on an image, and could rotate the observer's location around the object to be viewed. The display of points in three-dimensional space on a two-dimensional screen requires the use of perspective projection techniques also involving matrix multiplication (11:127-129). Kelso used such projection techniques in SAT-MAP2 (7).

Orbital Mechanics Algorithms. Algorithms have been developed to answer particular questions and to account for perturbations in the motion of satellites. Lawton described an algorithm to determine whether a satellite was visible to a particular ground station (8:32). Other algorithms account for the various perturbations affecting satellite motion.

> Two-body motion is the principal orbit motion. The orbits of most of the bodies in space can be described as two-body orbits to a fair degree of accuracy. (13:275)

However, Tang found that perturbations such as the tri-axial ellipsoidal shape of the earth could change satellite pass durations by up to a few minutes (14:447). State-of-the-art orbit determination, such as that done at NASA's Goddard Space Flight Center, requires a model which includes "a spherical harmonical representation for Earth gravitation, models for solar radiation pressure, atmospheric drag, and dynamical Earth and ocean tides (12:198)." SAT-MAP2, which will be used as a starting point for this project, used a procedure called SGP to calculate the motion of satellites. The procedure, based on the NORAD simplified general perturbation model, included basic two-body motion modified for atmospheric drag and first-order gravitational perturbations (7).

Summary.

The preceding paragraphs reviewed the literature for information relating to this project. They covered the following areas: 1) the need for such a model; 2) the design of user interfaces; and 3) graphics display and orbital mechanics algorithms. In addition to reviewing journals and texts, they included a review of similar computer programs and their code for information to use. There is evidence to believe a tool allowing the real-time manipulation of orbits in three dimensions would be useful in instruction and analysis. Such a tool would be especially useful if it used the powerful personal computers now widely available. The design of a user interface for such a tool requires careful consideration as to the interface organization, information to be displayed, and portability of the program. Many of the graphics display and orbital mechanics algorithms required for such a tool have already been developed, including algorithms for geometric transformations and orbit propagation.

III. Methodology

The chapter which follows describes the methodology used in developing SATMAP3 as an instructional and analytic tool from its predecessor SAT-MAP2 (7). It addresses the method used to select capabilities and features to be included in SATMAP3 and describes the decisions which went into designing the user interface for the program. It also describes the orbit propagation, transformation, projection, and scaling algorithms developed for the program.

Capability Selection.

The first task faced in developing SATMAP3 was the selection of which capabilities and features to include in the program. Discussions with the author of SAT-MAP2, Dr. T.S. Kelso, revealed many capabilities and routines which could be used or modified from that program (6). Such discussions also identified capabilities which did not exist in SAT-MAP2, but which would be needed in a tool designed for use by instructors and analysts. In particular, such a tool would need a much more extensive and easy-to-use user interface. An extensive review of existing literature and other computer programs revealed other possible capabilities and features.

Capability Selection List. Those features and capabilities of possible use in an instructional and analytic tool were organized into a list, and rated as to usefulness and difficulty of implementation on a scale of zero to nine. This list can be found in Table 1. The author then selected those capabilities to be included in the program which together accomplished the following: 1) resulted in a complete program; 2) avoided difficulties where there was another acceptable alternative; 3) added useful features where the difficulty involved was small; and 4) eliminated capabilities which substantially duplicated other features while adding little value. Further discussion with the thesis committee finalized this list of capabilities to be included in SATMAP3.

User Interface Implementation. A user interface involving menus with selections made using the arrow and enter keys was chosen because it is easy to learn and use. It offers great flexibility in setting up a scenario to be run, since only those options which must be changed need be selected. Selection by use of a mouse was discarded since one is not always available, and would still require use of the keyboard for data entry. Selection by use of speed keys was discarded as more difficult to learn, only useful to those few users who would be using the program frequently enough to memorize the speed keys, and, most importantly, redundant with the selected method.

TABLE 1

CAPABILITY SELECTION LIST
(RATING 0-9, 9 USEFUL OR DIFFICULT)

ITEM	USEFULNESS	DIFFICULTY	IMPLEMENT
USER INTERFACE IMPLEMENTATION			
Menu	9	8	Yes
Enter Select	9	4	Yes
Mouse Select	6	4	No
Speed Key Select	6	5	No
CONFIGURATION OF THE SCENARIO			
System Configuration	3	9	No
Observer (location, frame)	9	3	Yes
Print Data to File	6	5	Yes
Print Graphics (screen print)	5	9	No
Constellation Selection	5	5	Yes
Orbit Selection	5	6	No
Defaults	8	3	Yes
On-line Help	3	6	No
Documentation on Disk	9	2	Yes
REAL-TIME MODIFICATION			
Real-time Orbit Change	6	7	No
Real-time Observer Change	8	5	Yes
DISPLAY OPTIONS			
Observer Location	3	3	No
Orbit Parameters	7	3	Yes
Satellite Location	6	5	Yes
Ground Track	8	6	Yes
Coverage	6	7	No
Orbit Plane	3	6	No
Point(s) on Earth (Lat/Long)	6	5	Yes
ADDITIONAL FEATURES			
Rotating Earth Map	7	9	Yes
Shading of Orbits	3	7	No
Maneuvers By Script	4	5	No
Sat-Sat Distance/ Closest Approach	5	5	No
DATABASES			
Sample of Satellites	9	1	Yes
Extensive Set of Satellites	6	9	No
Orbit Parameters	9	1	Yes
Orbits	9	3	Yes
Space Systems	8	3	Yes
Perturbations	6	5	No

Configuration of the Scenario. Significant decisions related to configuration of the scenario included the method of configuration to match hardware, the use of defaults, and the method of providing documentation or help. Of particular importance in the program design was the decision to keep the program usable by all hardware configurations without the need for a separate hardware configuration step or the use of separate printer drivers to support screen prints. Screen prints, while of some use, were avoided as too difficult to implement because a different printer driver would be required for each printer supported. Further, screen prints would not be accurate enough for many analytic applications. Instead, a routine to allow position information of three types to be dumped to ASCII files for later analysis by hand or other user-owned software was developed. The inclusion of default values, updated to the last value used, and saved for future use, was critical to making the program easy to use. Combined with menu selections, this allowed scenarios to be quickly modified and re-run, changing only a few parameters each time. Program documentation was put on disk so it would be always available with the program. This was much easier to implement and more complete than having an on-line help feature.

Real-Time Modification. Real-time modification (during scenario execution) of the observer location with Pan and Zoom features was included as useful in studying the details of particular situations. Real-time modification of the satellite orbit was not included because it was difficult to implement with the way satellite parameters were being stored in external files. Investigation of the effects of such changes could instead be done by re-running the program and picking a new satellite or constellation file each time.

Display Options. A number of features were selected which affected information displayed on the screen during scenario execution. The display of orbit parameters was particularly useful during instruction. An orbit parameter and satellite location display would also be useful in providing a quick look analysis of the situation and in data collection. Similarly, display of a ground track would be useful in instruction and in a quick check of the satellite's position relative to the earth. The display of user defined points on the earth was a particularly innovative and important feature. This allowed much quicker execution than displaying a map, was more accurate in locating a particular point of interest than any map, and was essential to File Output Format 3, which outputs look angles from a point on the earth to a satellite. A feature to show earth coverage by a satellite was discarded for two reasons. Coverage varies from satellite program to program, with some allowing coverage of all areas visible, others requiring some minimum elevation angle, and others having a much more restricted swath. Similar results can often be obtained by placing the observer on the satellite and checking for visibility of various earth points or map locations, by observing the location of the ground track, or by analyzing the position numbers output to a file (File Format 2: latitude, longitude, height; or Format 3: look angles, are particularly suited to this). Finally, display of the orbital plane was discarded since the same result could be obtained by properly selecting the time interval between position calculations and observing the past locations of the satellite through one revolution. Overall, the wide selection of display options offers great flexibility and application if used with some imagination.

Additional Features. The addition of other features was limited due to the thesis time constraints and the need to develop a completely new user interface. However, to really handle all of the desired observer reference frames, a capability to display a rotating grid or map was needed. SAT-MAP2 only displayed a grid or map if the observer was rotating with the earth, hence the map could be drawn once and not updated during execution. This was a difficult feature to add, in that it required calculation and drawing of the screen at each time increment. However, the inclusion of a grid and map allows a quick look understanding of the situation, and was one of the original reasons for the project.

Databases. The selection of which satellite and constellation element set files to include in the database would be critical in making the program easy to use for instruction, but less so for use in analysis. An analyst would need element sets for a particular time (epoch) and should have access to such data. It was decided to include a good-sized sample of satellites, but not try to be all-inclusive or up-to-date, since this would be an impossible task. Rather, the program is designed to accept a standard NORAD element set in an external file, so the user can build his own files without the need to modify or compile the program. For ease in instructional use, database files which could be used to illustrate the various orbital parameters, types of orbits, and example space systems were included. Illustrations of various perturbations were left to the instructor to develop, as it was felt these would be less useful and also would be more difficult to predict which to include.

User Interface.

The user interface for SATMAP3 consists of a main menu, three submenus, and a number of data input blocks for user input, and the output of data to the screen and external files.

Input. Six generic menu drawing procedures and 16 procedures specific to particular menu selections provide for menu selection and parameter input.

Menu Selection. The menus are organized into one main menu, which feeds three submenus (one each for set up, display options, and execution), each of which connects to several procedures for parameter input. The menus are organized in the typical sequence in which they would be used during a run, with the first on top and exit selections on the bottom. The main menu is displayed on the screen top left, submenus on the display bottom left, and parameter input blocks pop up on the right, where the satellite and earth display are during execution. All displays are in EGA graphics format, for ease in performing memory screen swapping, as discussed later.

Menu selections are made with a system in which the current selection is highlighted in green, the current selection is moved from item to item with the up and down arrow keys, and an item is selected with the enter key. Scrolling of selections does not wrap around, top to bottom, for ease in reaching the exit selections; the user can just hold the down arrow key down. The current menu selection is kept track of with menu and submenu index variables. These are incremented and decremented as the up and down arrow keys are pressed. When a submenu selection is finally made, a new block is displayed for parameter input.

Parameter Input. The input of various parameters (such as observer frames and locations, file names, which display options to use) is performed on separate pop-up blocks on the right of the screen. A generic procedure draws the block outline and displays input prompts and default values. Additional messages describe the proper format and the meaning of the various parameters. The various input fields are edited one at a time, character by character. This allows for easy editing since the current values remain visible, and only the required characters need to be changed. When a field is finished, the enter key is pressed to go on to the next field. When all fields are done, the input strings are converted to their proper format, such as double precision real numbers, integers, or date-time strings. Limited format checking is done, and if errors result the user is given an error message and the input block is again displayed. Any other calculations peculiar to the selection are done, the defaults are updated, and the user is returned to the submenu. All menus have exit selections so it is easy to get out of the program.

Output. Program output consists of a screen display (of the earth and the satellites revolving around it) and possible data dumps to external files in ASCII format. Output is displayed on the right side of the screen in EGA graphics format. Calculations and drawing to the screen are done on one of two screen memory pages while the other page is actually displayed on the screen. Then the screen memory pages are swapped, so the new information is displayed on the screen and the old page can be overwritten. The result is a much more crisp change from one time increment to the next, rather than watching the screen be erased and drawn again, pixel by pixel. EGA graphics format is used since it is readily available and screen memory paging has peculiar problems in higher resolution formats. Menu selections allow turning various displays on or off to avoid cluttering the screen with unneeded data and to speed program execution.

The file output feature should be particularly useful to analysts. It allows satellite position data of one of three types (rectangular ECI coordinates; latitude, longitude, and height; or ground-point-to-satellite look angles) to be output to an external file in ASCII format for later analysis. Such a file can be edited and then displayed with a graphics program or input into a user-written computer program to do project-specific calculations. The difficult calculations of figuring out where the satellite will be compared to inertial space, the rotating earth, or some particular earth point will have already been done by SATMAP3.

Description of Algorithms.

SATMAP3 uses orbit propagation, transformation, projection, and/or scaling algorithms to perform calculations on ten different types of elements which together make up the screen and file output of the program. Elements which require some or all of these calculations include the following: 1) the observer location; 2) a latitude, longitude grid; 3) an outline of the earth horizon; 4) an earth map; 5) various earth points defined by the user; 6) latitude and longitude coordinates for File Output Format 2; 7) look angle coordinates for File Output Format 3; 8) old satellite positions; 9) current satellite positions; and 10) the satellite ground track. While each of these elements requires slightly different processing, many of the algorithms can be used as part of the processing for more than one element. Therefore SATMAP3 is organized into a number of different modules or procedures, with the flow between them controlled during execution by Procedure Select_Start. This section describes the various algorithms used to perform orbit propagation, transformation, projection, and scaling which are controlled by Select_Start, and the following section describes the complex interactions between these algorithms.

Select_Start. Procedure Select_Start provides overall control of the program during scenario execution. As described earlier under User Interfaces, it manages the two screen memory pages available in EGA graphics format, displays one page while it writes to the other, and swaps them after each time increment. It also contains a section to check for interrupts. During execution, the normal processing is suspended when a keyboard key is pressed. This allows the program to be paused or the observer location modified with the Pan and Zoom commands. During each time increment, Select_Start controls the processing required by the ten elements mentioned previously. It performs some processing itself and invokes other algorithms as required for each element. Select_Start keeps track of the current time, advances it one time increment (user definable) after all processing at the current time is complete, and ends the scenario run when the stop time is reached.

Draw_Grid, Draw_Earth, Draw_Map. These three procedures coordinate the drawing of a latitude/longitude grid, the earth horizon, and an earth map, respectively. Draw_Grid draws circles of equal latitude every 20 degrees from -80 to +80 degrees north latitude and circles of equal longitude every 30 degrees. The routine used in SAT-MAP2 drew these circles by plotting points every few degrees. The routine used in SATMAP3 is modified to draw lines between points rather than the points themselves. This allows fewer points to be used, speeding execution. Draw_Earth has been left unchanged from SAT-MAP2, and draws lines between points incremented around the outside of the circle which forms the earth's horizon. Before Draw_Map can proceed with drawing an earth map, it first has to read in the map coordinates from an external file. The routine is essentially unchanged from that used in SAT-MAP2. Map coordinates are stored as longitude, colatitude pairs in a record format in the file, with a coding on the longitude coordinate to indicate whether the coordinate is the start or a continuation of a line to be drawn. SAT-MAP2 only drew an earth map in a reference frame rotating with the earth, where the map could be drawn once per scenario and left on the screen throughout the scenario run. To handle other reference frames, SATMAP3 has to draw the map again each time increment as the earth rotates with respect to the observer. This procedure slows program execution down badly compared to running the program without a map.

To partially alleviate this problem, a new, smaller map file was developed for SATMAP3. This effort started with a file used by a BASIC program provided by Major Bruce Morlan. The file was converted to a format readable by Pascal, and the mixed integer, real number contents converted to all real numbers. The file's latitude coordinates were converted to colatitude, and the longitude converted from west longitude to east longitude. Then the precision of the numbers was converted to a constant tenth of a degree precision, and they were stored as tenths of a degree vice degrees. Finally, the existing coding for start or continuation of a line was stripped off and replaced with that coding expected by SATMAP3. The result is a file completely compatible with Draw_Map, with far fewer points, which provides ample detail but allows the program to execute 6.4 times faster than using the old map file.

Update_Observer_Location. Procedure Update_Observer_Location propagates the location of the observer, in earth-centered inertial (ECI) rectangular coordinates, to its new location at the new time. For reference frames -1 (ECI) and 0 (rotating with the earth) this is accomplished by converting the observer latitude, longitude, and height into rectangular coordinates, and rotating these about the spin axis by the proper angle to bring the x axis in line with the first point of Aries, hence earth-centered inertial coordinates. For reference frames on a satellite, Procedure SGP is used to derive the new ECI coordinates.

LatLong_to_ECI. Procedure LatLong_to_ECI converts latitude and longitude coordinates with respect to the rotating earth to ECI coordinates. This is performed by converting latitude and longitude to rectangular coordinates on the surface of a spherical earth, and then using a simple matrix multiplication transformation to rotate the coordinates to earth-centered inertial ones.

SGP. Procedure SGP calculates a satellite's position and velocity vectors in ECI coordinates from NORAD two-line element set (ELSET) data stored in an external file and the time since the epoch of that data. This algorithm was developed by Hilton and Kuhlman in 1966 and appeared in Hoots (4) along with documentation. The procedure used in SATMAP3 was taken directly from the Pascal procedure used in SAT-MAP2 (7). SGP uses a simple linear drag model and takes into account some of the perturbations caused by a non-spherical earth. The algorithm was originally designed for use with near-earth satellites, but because of the way NORAD element sets are generated with a pseudo-drag term included, it can be used for either near-earth or deep-space orbits (4:2).

Project_Point. Procedure Project_Point uses a perspective projection to project a point in earth-centered inertial (ECI) coordinates onto a projection plane with x and y screen coordinates and with respect to an observer at some finite distance from the earth. The projection plane used is perpendicular to the vector from the center of the earth to the observer. It slices through the earth in the circle formed by the earth's horizon as viewed from the observer. The z axis of the projection plane is out of the plane, toward the observer. The y axis is in the plane formed by the earth-observer vector and the spin vector of the earth. The x axis is positioned in the projection plane to form a right-handed, rectangular coordinate system. In simpler terms, when viewed by the observer, the x and y axes are in the projection plane, with the y axis generally northward and the x axis 90 degrees to the right. Once the point location, observer location, and projection plane are specified, the projection of the point is found using the similar triangle geometry of the situation as done by Plastock

(11:128,129) and the projection is transformed into ECI coordinates using three-dimensional rotations, also described by Plastock (11:116-118).

Draw_Line, SetPoint, Scale_Point. These three procedures together convert a point in the projection plane with x and y coordinates in kilometers to pixel coordinates and either plot the point (SetPoint) or draw a line between two such points (Draw_Line). Procedure Scale_Point does the actual scaling from the projection plane coordinates to pixel coordinates with translation and reflection of axis transformations. The projection plane coordinates are in kilometers, with the origin at the center, +y up and +x to the right. The final pixel coordinates have the origin at the top left of that portion of the screen used by the satellite display, with +y down and +x to the right. As described in more detail in the Change Magnification section of Appendix A, with a magnification setting of 1.0, the size of the image on the screen is meant to approximate the same angular size that would be seen by the observer.

Interaction of Algorithms.

The algorithms just described, controlled by Procedure Select_Start, perform the necessary propagation, transformation, projection, and scaling calculations needed to draw on the screen or write to a file the ten elements which follow: 1) the observer location; 2) a latitude, longitude grid; 3) an outline of the earth horizon; 4) an earth map; 5) various earth points defined by the user; 6) latitude and longitude coordinates for File Output Format 2; 7) look angle coordinates for File Output Format 3; 8) old satellite positions; 9) current satellite positions; and 10) the satellite ground track. A quick description of the interactions of the above algorithms to deal with each of these elements now follows. These interactions are also illustrated in a flow chart in Figure 1.

The Observer Location. Procedure Select_Start controls calculation of the observer location by invoking Procedure Update_Observer_Location at the start of each time increment. This procedure performs internal calculations to find the observer location in ECI coordinates when the observer reference frame is earth-centered inertial or rotating with the earth. When the observer reference frame is traveling on a satellite, Procedure Update_Observer_Location invokes Procedure SGP to calculate the observer location from the element set of the observer satellite and the current time. In either case, the result is an earth-to-observer position vector, x0, a global variable that can be accessed by other procedures in their calculations.

A Latitude, Longitude Grid. Procedure Select_Start invokes Procedure Draw_Grid to draw a latitude, longitude grid each time increment. This is only done in a scenario run that the parameter Grid is set to ON using the Display Options submenu. From this point, Procedure Draw_Grid

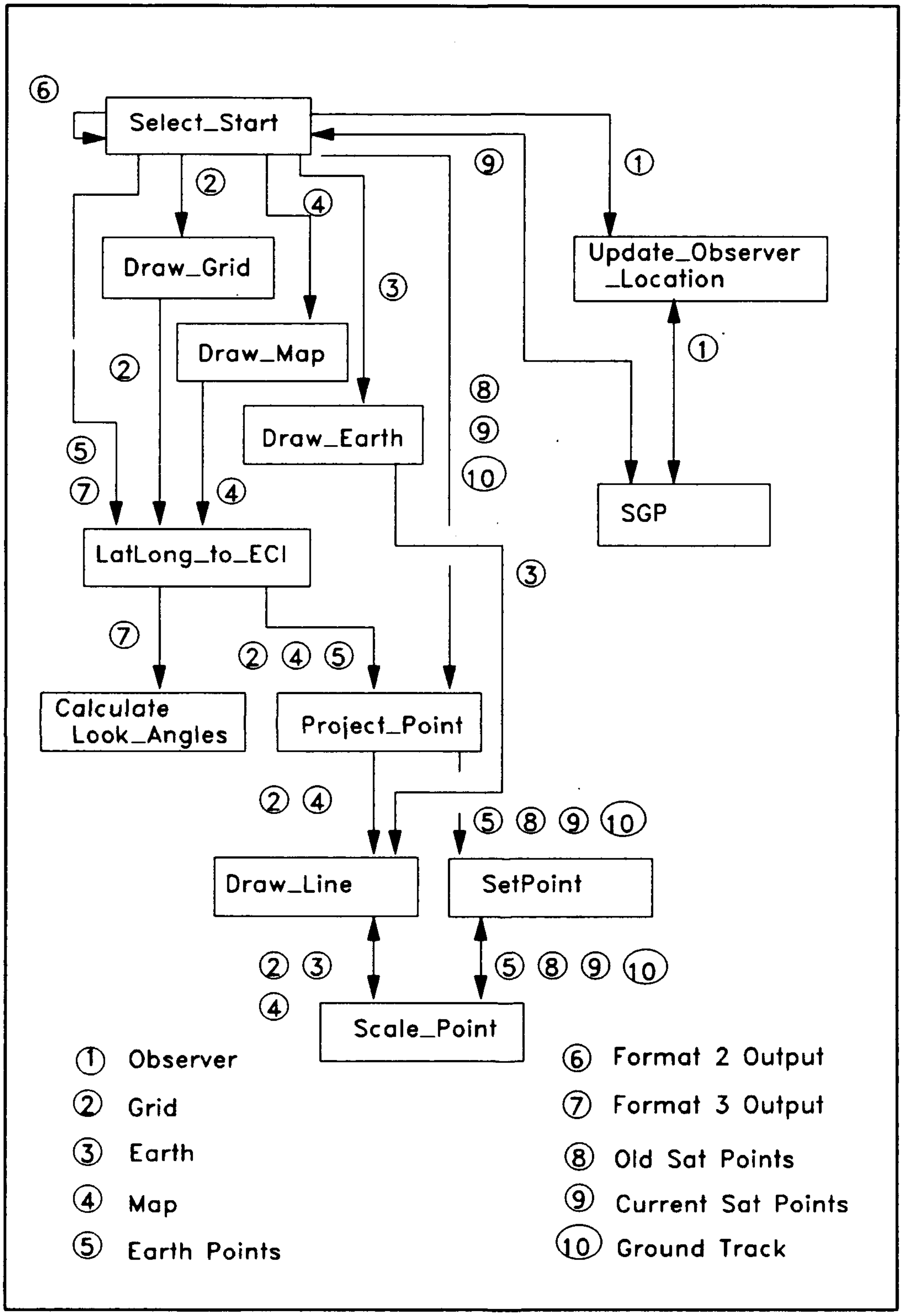

Figure 1. Algorithm Flowchart

coordinates actions to draw the grid. It invokes Procedure LatLong_to_ECI multiple times to convert earth latitude, longitude coordinates to ECI ones. It then uses Procedure Project_Point to project them onto the projection plane. If the point is visible, it uses Procedures Draw_Line and Scale_Point to actually draw the grid on the screen.

An Outline of the Earth Horizon. Procedure Select_Start invokes Procedure Draw_Earth to draw a circular earth horizon every time increment of every scenario. Since the earth drawn is spherical, there is no need to propagate or transform points to do this. Rather, Procedure Draw_Earth starts by projecting points on the earth's horizon onto the projection plane and drawing them using Procedures Project_Point, Draw_Line, and Scale_Point.

An Earth Map. Procedure Select_Start invokes Procedure Draw_Map to draw an earth map each time increment. This is only done in a scenario run that the parameter Map is set to ON using the Display Options submenu. From this point, Procedure Draw_Map coordinates actions to draw the Map. It reads in coordinates from the external map file and determines whether each point is the start or continuation of a line on the map. It invokes Procedures LatLong_to_ECI, Project_Point, Draw_Line, and Scale_Point in the same manner that Draw_Grid does to draw the earth map.

Various Earth Points. Procedure Select_Start directly coordinates the drawing of points on the earth which are specified on the Display Options submenu prior to execution. Since these are just latitude longitude coordinates like those used in drawing the grid and map, they require a similar sequence of procedures: LatLong_to_ECI, Project_Point, SetPoint (vice Draw_line used by map and grid, since only drawing one unconnected point), and Scale Point.

Latitude, Longitude Coordinates for File Output Format 2. Procedure Select_Start handles this calculation directly. It contains an algorithm to convert a satellite's ECI rectangular coordinates into latitude, longitude, and height coordinates by first rotating the ECI coordinates into a frame fixed to a rotating earth with the new x axis at 0 degrees longitude, and then converting the rectangular coordinates to angles using trigonometry. Finally, it writes these values to the external output file.

Look Angle Coordinates. The calculation of look angle coordinates (azimuth, elevation, slant range) used in File Output Format 3 is coordinated by Procedure Select_Start. It invokes Procedure LatLong_to_ECI to convert the earth point used as the origin of the new topocentric reference frame to ECI coordinates. Select_Start then invokes Procedure Calculate_Look_Angles. This procedure calculates the look angle coordinates from an earth-center-to-satellite vector and an earth-center-to-earth-point vector, both in ECI coordinates. Select_Start then writes these coordinates to the output file.

Old Satellite Positions. Procedure Select_Start controls the drawing of old satellite positions. These positions are stored in a global three-dimensional array which indexes satellite number, point number for that satellite, and x, y, or z coordinate for that point. Memory limitations restrict the size of the array used, so only the last 50 positions for each satellite are stored and drawn. This array is updated with the ECI coordinates of each satellite as the new satellite position is calculated, so there is no need to use Procedure SGP to find the coordinates of the old points. Select_Start does invoke Procedures Project_Point, SetPoint, and Scale Point to project and display these old satellite positions.

Current Satellite Positions. Procedure Select_Start first uses Procedure SGP to calculate the ECI coordinates of the satellites at each time increment. It then uses Procedures Project_Point, SetPoint, and Scale_Point to project and draw these positions. At the same time, it stores the current satellite position in the array used for old satellite positions, and shifts each old satellite position in the array so only the most recent 50 positions for each satellite are stored.

The Satellite Ground Track. The last element controlled by Procedure Select_Start is the drawing of a satellite ground track for one satellite as selected in the Display Options submenu. After calculating the selected satellite's position for the element above, the same position is used to find the current ground track position. Each of the ECI coordinates is divided by the magnitude of the earth-center-to-satellite vector, and multiplied by the radius of the earth. The result is a vector pointing toward the satellite, but of length one earth radius, hence the position of the ground track on a spherical earth. This position is stored in a two-dimensional array indexing the number of the ground track point (such as the first point, second) and the rectangular coordinates of the point. The history of the ground track consists of the points on the earth that the satellite was over in the past, not the ECI coordinates of these points. In other words, the old ground track points have to rotate with the earth, so as to remain at a constant point on the surface of the earth, not at constant ECI coordinates. Since the angle rotated by the earth each time increment is fixed throughout the scenario, the current locations of the complete ground track can be obtained from the locations at the last time increment by rotating all of the points by this rotation angle. The array stores the coordinates of each ground track point at its current ECI coordinates, not at the ECI coordinates of the point when the satellite was over that point. Given this array of ECI coordinates of the ground track at the current time, Procedure Select_Start then uses Procedures Project_Point, SetPoint, and Scale_Point to project and draw the ground track points. The storage array is updated to include only the most recent 100 ground track points.

Summary.

The preceding chapter described the methodology used in developing SATMAP3 as an instructional and analytic tool from its predecessor SAT-MAP2 (7). It addressed the method used to select capabilities and features to be included in SATMAP3 and described the decisions which went into designing the user interface for the program. It provided a description of the various algorithms used for orbit propagation, transformation, projection, and scaling. It then described how the interaction of those algorithms is controlled by Procedure Select_Start to generate the program output.

IV. Validation

Validation of SATMAP3 was performed in three parts. Critical algorithms used were validated separately and the program as a whole validated, as well. The user interface was validated and tested to ensure the proper responses to selected actions occurred. Finally, user testing was done to obtain feedback, validate the user's guide, and check program compatibility on various computer systems.

Algorithm Validation.

Three critical algorithms and their associated subroutines were validated separately and SATMAP3 validated as a whole in a number of tests. Procedure Update_Observer_Location was first validated to ensure the proper conversion of observer latitude, longitude, height coordinates to earth-centered inertial (ECI) coordinates, and the proper propagation of those coordinates through time in each of the various reference frames. The reference frames fixed to a satellite also validated the linkage with Procedure SGP that is used to propagate the satellite's position through time. Validation of Procedure Project_Point then showed that observer and satellite ECI coordinates are properly projected to projection plane coordinates, and scaled to screen pixel coordinates. A separate validation effort of Procedure Calculate_Tsince was also done, since this procedure was changed to correct a known bug from SAT-MAP2 concerning calculation of the time elapsed since the satellite epoch (variable tsince in the program) when the epoch and scenario start time span one or more years. The program as a whole was then validated in three separate tests.

Validation of Update Observer Location. A series of nine test cases were run. In each, an observer latitude, longitude, and height (or satellite element set, for observers on a satellite) were provided, and the resultant ECI position, latitude, and longitude coordinates at various future times was output. Test cases were chosen in which the results could be predicted using simple orbital mechanics. For example, an observer in an ECI frame shows constant ECI rectangular coordinates and constant latitude, but a longitude which decreases by approximately 15 degrees per hour (360 degrees per day) as the earth rotates beneath the observer. An observer rotating with the earth shows varying ECI coordinates, but constant latitude and longitude. An observer riding on various GPS satellites (12-hour, semi-synchronous orbits) shows proper altitude, varying ECI coordinates with constant magnitude and about a 12-hour period, latitudes which vary with a 12-hour period and reach a maximum equal to the orbital inclination, and longitudes which vary with a 24-hour period (plus 1 degree, or 4 minutes, per day, as proper for those satellites). Tests using an observer on geosynchronous GOES satellites show, most importantly, a constant longitude. They also show proper altitude, ECI coordinates of constant magnitude but values which vary with a 24-hour period, and a latitude close to zero, but which vary slightly with a 24-hour period. The result of these tests is to validate that the program is accurately converting and propagating observer and satellite locations into ECI coordinates.

Validation of Project_Point. A validation of Procedure Project_Point was done to show that observer and satellite ECI coordinates are properly projected to projection plane coordinates, and scaled to screen pixel coordinates. In a series of 16 test cases, the ECI rectangular coordinates of the observer and the point to be projected were input, and the resultant coordinates in the projection plane, visibility flag, and screen pixel coordinates checked. Test cases were chosen so the results could be verified with manual calculations, and checked using a variety of observer to point geometries. Projection plane coordinates are calculated by the program within a kilometer of those calculated by hand, and screen pixel coordinates are calculated to the nearest pixel. These tests validate the projection and scaling algorithms used by SATMAP3.

Validation of Calculate_Tsince. In order to validate this algorithm, a program was developed which inputs a scenario start time and element set epoch time, and using Procedure Calculate_Tsince, can calculate the time between the two in minutes. A total of 54 test cases were run to systematically test the possible relationships between start time, epoch, the end of the year, leap days, and lack thereof. The routine is not designed to predict leap seconds, nor to be accurate beyond the year 2099. In other respects, the program accurately calculates the time as verified by comparison with calculations performed by hand. These tests validate that the times being used by the propagation algorithms are accurate regardless of the relationship between the start time and the epoch.

Validation of Algorithms Combined. Three separate tests were used to validate the propagation, transformation, projection, and scaling algorithms as a whole. The first test used an element set for a NOAA satellite to compare the ECI coordinates of the satellite at a time after the epoch with the coordinates expected from other calculations (5). Using the element set contained in the file VALSAT.TLE included with the program, SATMAP3 calculates ECI coordinates at 19:39 on 30 Jan 1991 of x = 5982.2169, y = -2191.1092, z = 3400.1993 km compared to predicted coordinates of x = 5982.217, y = -2191.109, z = 3400.199 km.

A second test was done using the test case from the documentation for Procedure SGP (4). Using the element set contained in the file SGPVALID.TLE, SATMAP3 calculates ECI coordinates at 23:41:24 on 1 Oct 1980 of x = 2328.63, y = -5995.10, z = 1720.78 km compared to the SGP documentation of x = 2328.96, y = -5995.22, z = 1719.98 km. At 05:41:24 on 2 Oct 1980, SATMAP3 calculates values of x = 2455.70, y = -6071.89, z = 1223.78 km, whereas the SGP documentation predicted values of x = 2456.00, y = -6071.94, z = 1222.95 km. In each of the above cases, SATMAP3 calculates coordinates within 1 kilometer of those predicted. Since NORAD element sets are accurate only to within about 12 kilometers, it is clear that SATMAP3 is using Procedure SGP and the other program algorithms correctly to calculate, display on the screen, and store in the output file a satellite's ECI coordinates.

A final test of the SATMAP3 algorithms together ran various scenarios and looked for various conditions which could be predicted beforehand. Satellites with circular orbits give circular displays when viewed from the correct perspective, while highly elliptical ones give elliptical displays. Orbits with zero inclination are displayed over the equator, while polar orbits orbit the poles. The ground track of a satellite reaches a maximum latitude equal to the inclination of its orbit. Satellites in the same orbit but with different mean anomalies stay in the proper phase relationship. Semi-synchronous satellites have 12-hour periods with a ground track that repeats every 24 hours. Geosynchronous satellites have 24-hour periods and ground tracks which are a point. The algorithms together give predictable results.

In conclusion, the efforts described above provide extensive validation for the propagation, transformation, projection, and scaling algorithms of SATMAP3. The program started with a well-documented algorithm, SGP, which was transferred without modification from SAT-MAP2. The manner in which Procedure Update_Observer_Location uses SGP to propagate observer and satellite locations was tested with positive results. The projection and scaling algorithms that Procedures Project_Point and Scale Point use were tested to ensure the ECI coordinates generated by SGP are properly projected and scaled to the screen display. Again, the results were positive. A test was done to ensure that start and epoch times are properly converted into the time since the epoch by Procedure Calculate_Tsince. Tests were run to compare the output ECI coordinates from SATMAP3 with those from other sources, with resultant accuracies better than 1 km. Finally, a number of tests were run to ensure SATMAP3 handles various scenarios to provide output which can be predicted beforehand. The result of this extensive validation effort is strong evidence that SATMAP3 accurately models satellite orbits.

User Interface Validation.

The user interface for SATMAP3 and the procedures used to implement it were tested as each procedure was developed to ensure each procedure gives the proper responses and results. This was much easier to test than the orbital mechanics algorithms above, since by its nature a user interface gives extensive feedback.

Menu selection and parameter input procedures provide direct feedback through the selections they highlighted and the parameter values they display.

Similarly, it was quickly apparent by running the program that setup selections which modify the constellation, observer frame, and observer location give the proper results. The Set Time selection was checked by comparing the time input with that displayed on the screen during execution, that output to the output file, and by stopping the program and examining the values of the various time variables. File output procedures were tested by comparing the values output to the file with the graphical and numerical results displayed on the screen. The Reset Defaults selection was checked by examining the contents of the default file and by running the program with and without various modifications and resets in between.

The various display options were also validated. It was especially useful to compare the output of one, such as the grid, with the output of another, such as a point on the earth. This allowed the graphical displays, for example Grid, Map, Ground Track, and Point on the Earth to be compared with each other, and with the orbit parameter and satellite location numerical displays.

The various execution selections were checked in the course of the preceding validation. Many tests required running the program with the Start selection. The Pan and Zoom features allowed examination of the above scenarios from multiple vantage points to more clearly see what was going on. At the same time this checked the proper workings of these features.

In summary, the user interface for SATMAP3 was validated primarily by running the program, testing each feature, observing the resulting feedback, and comparing the results from one selection with the same information shown by another selection. When information is input in the proper format, and the appropriate menu selections are made, the program gives the correct output.

User and Compatibility Testing.

User testing was performed to obtain feedback, validate the user's guide, and check program compatibility on various computer systems. Testing was performed by the author, the two members of the thesis committee, and by two students in the Graduate Space Operations program at the Air Force Institute of Technology (AFIT) on a total of seven computer systems. Extensive feedback from the thesis committee was incorporated into the program and user's guide. Subsequent testing by AFIT students confirmed that the modified program and documentation were effective. No compatibility problems were experienced on any of the computer systems used, but the program did run noticeably slower on a machine with a 286 processor compared to a 386SX processor, and slower using a floppy disk drive compared to a hard disk drive. The validation efforts just described support a conclusion that the program works, that the documentation is sufficient, and that the program is compatible with a wide variety of IBM-compatible personal computers.

Summary.

An extensive effort was made to validate SATMAP3 as a computer graphics model of satellite orbits for use in instruction and analysis. The propagation, transformation, projection, and scaling algorithms were validated in logical subsets and as a whole by comparison with hand calculations and with output from other sources. The user interface was validated through the direct feedback of examining the screen display, by comparison with similar information displayed in another manner, and in some cases by examining the values of variables when the program was stopped in the middle of execution. User and compatibility testing was done to ensure that the user's guide is effective, that the program is compatible with a variety of IBM-compatible computer systems, and to solicit feedback on the implementation of the program. The results of these efforts support a conclusion that the program is valid for its intended purpose.

V. Results

The result of this research effort was the development of a completed computer graphics model of satellite orbits, suitable for use in instruction and analysis. This includes a working computer program, SATMAP3, a user's guide, a set of sample satellite databases, suggestions on how to use the program in instruction and analysis, and documentation of the effort. The effort started with a program called SAT-MAP2, which consisted of 25 procedures, 20 functions, and 1200 lines of code. The source code occupied 34 kilobytes of memory, and the executable version 53 kilobytes. Building upon this as a base, SATMAP3 was developed and contains 49 procedures, 19 functions, and over 3200 lines of code. The source code occupies 109 kilobytes and the executable version 82 kilobytes of memory. All 19 functions are copied from SAT-MAP2 with only cosmetic changes. Of the 49 procedures in the new program, 9 are copied as is from the earlier one with only cosmetic changes, 7 are copied and significantly modified, and 33 of the procedures are new. In order to facilitate its use as an instructional and analytic tool, SATMAP3 has a menu driven user interface. Many features were also added to the old program to make it more suited for instructional and analytic use.

SATMAP3 displays many types of information on the screen concerning satellite orbits; it also has a completely new feature allowing it to output information to an output file. The observer reference frame can be selected as either inertial, rotating with the earth, or on one of the satellites. The program displays the current locations of a constellation of up to 50 satellites, and up to 50 of the last locations of each satellite. It can plot a ground track for any one of the satellites in the constellation. It can display a latitude, longitude grid and an earth map regardless of the reference frame used. It can display up to 15 user defined points on the earth. This was an important new feature which allows the user to see the location of ground stations, targets, or other points of interest far more accurately and quickly than by using a map. Satellite location data can be output to a file in ASCII format using one of three types of coordinates (rectangular earth-centered inertial; latitude, longitude, and height; or look angle azimuth, elevation, and slant range). It can also display on the screen during execution the orbit parameters or satellite location and velocity of any one satellite. The program displays the image in the apparent angular size that it would be if seen by an observer at the observer location, and this can be modified by a magnification factor. During execution of the scenario, the program can be paused, and the observer location modified by panning and zooming around the scene. Throughout the menu selection process, all parameters have defaults, these defaults are dynamically updated with new input values, and stored for use as the new default values the next time the program is run.

An extensive validation effort was undertaken to ensure the program works properly, that the user's guide is effective, and that the program works on a variety of IBM-compatible computers. In particular, the program accurately propagates the satellite location from a NORAD two-line element set to within 1 kilometer of the location predicted by other sources, when a current satellite epoch is used.

This research resulted in the development of SATMAP3, a dynamic computer graphics model of satellite orbits, suitable for use in instruction and analysis. A user's guide and some suggestions on how to use to the model in instruction and analysis are contained in appendices to this document. It is hoped the model will prove useful to satellite operations instructors and analysts both at the Air Force Institute of Technology and elsewhere.

VI. Recommendations

After completing this research project, three recommendations are made as to its use, distribution, and concerning areas for possible future research.

The first recommendation concerns the use of the program. SATMAP3 should be used in teaching orbital mechanics and satellite operations wherever access to an IBM-compatible personal computer makes its use possible. In teaching orbital mechanics, its use brings the dynamic nature of satellite orbits to life as no paper drawing can. When used to teach satellite operations, SATMAP3 adds a valuable perspective on the nature of various orbits that is easily remembered by the student. In analysis, SATMAP3 should be used where its convenience, availability, and simplicity of use are more important than the greater accuracy and speed available on some mainframe computers.

The second recommendation made concerns the distribution of the program. The author is a firm believer that knowledge spread widely is less likely to be lost, and have to be developed again. For this reason the author encourages the Operational Sciences department at the Air Force Institute of Technology, and anyone else who may obtain a copy of the program, to make it (including the source code) available to anyone to whom it may be of use. Modifications may be made to the program but should not be distributed other than by the authors. The program itself can be obtained from:

AFIT/ENS
Wright-Patterson AFB, OH 45433-6583
Attn: GSO Program Director
DSN 785-3362

The final recommendation concerns areas of possible future research. The capability selection list in the methodology chapter lists some features which were considered for inclusion, but left out due to time limitations. Of these, two would be perhaps of significant use. Some method of producing a screen print or screen image would be useful, but was not included due to difficulties with the wide number of printers available, each of which requires a separate printer driver. Someone with more knowledge of printers than the author might be able to add such a feature, either as a direct screen print or as a post processor acting on an output file. A feature which might be of use to both analysts and instructors would be a capability to display a satellite undertaking an orbital maneuver by some planned script. This would involve changing the orbital parameters at some time during execution, either switching satellite element set files or calculating a new element set and starting to use it after the maneuver. A feature which was not considered for inclusion, but which might be of use in some specific analyses would be the capability to model the attitude and position of the satellite compared to the sun.

Such a capability would allow study of the power received by the solar arrays with varying sun-vehicle-earth angles and times during the orbit. It could also allow the study of earth eclipses with respect to the satellite. Since SATMAP3 generates output files in ASCII format, another project might involve the generation of a post processor, written in any language, which could use the files output by SATMAP3 as an input to conduct further, and perhaps project specific, analysis. The automated calculation of satellite to satellite distance, distance and time of closest approach, or the analysis of coverage provided all might benefit by such a post processor. It is hoped that such recommendations may lead to further additions to the program, or suggest areas of possible future research.

Appendix A: User's Guide

User's Guide To SATMAP3

A Dynamic Computer Graphics Model of Satellite Orbits

For Use in Instruction and Analysis

SATMAP3 by John Anton, 14 October 1991

Based on SAT-MAP2 by TS Kelso, 12 November 1989

Appendix A: User's Guide To SATMAP3

This appendix is designed to provide the user of SATMAP3 with information needed to install and run the program.

Installation.

Hardware requirements to run SATMAP3 include an IBM-compatible personal computer with a 286 or better processor, an EGA graphics card and monitor, and 1Mbyte RAM. SATMAP3 runs more quickly from a hard disk, but can also be run from a single floppy disk.

Hard Disk Installation. Copy all of the files on the floppy disk to a hard disk directory called \TP\SATMAP3\. For example, from the A: floppy drive to the C: hard drive use the command:

```
XCOPY A:\TP\SATMAP3\*.* C:\TP\SATMAP3\*.*
```

Floppy Disk Installation. The program and related files must be in a directory called \TP\SATMAP3\. If they are not, copy all of the files on the floppy disk into a directory called \TP\SATMAP3\.

Execution.

To run the program, change the current directory to the one with the program, and execute it. If the program were stored in a directory called C:\TP\SATMAP3, then use the command:

```
CD C:\TP\SATMAP3
```

followed by the command:

SATMAP3

to start the program. After a few seconds for initialization, a MAIN MENU screen should be displayed.

<u>Menu Selection and Data Input</u>. The following keys are active in the program:

Up and down arrows	- scroll through menu
ENTER	- selects a menu item or enters a data item
Right and left arrows	- position cursor within a data field for editing
Backspace	- same as left arrow
A-Z, a-z	- input text in uppercase
0-9, + , - , .	- input number or symbol
\ , : , .	- input symbol used in file names
Spacebar	- blanks out a character
<ESC> or Control-X	- returns field to value before editing

<u>Main Menu Selections</u>. There are four selections on the MAIN MENU screen. The Set Up selection connects to submenus controlling the orbital scenario to be analyzed by the program. The Display Options selection connects to submenus controlling how the program output will appear on the screen. The Execution selection connects to submenus allowing execution of the scenario and limited real-time modifications to the scenario. Finally, the Exit Program selection will end execution of the program. All input parameters have defaults, and the program will store the last values or settings used as new default values.

Since all parameters are defaulted, a typical program run would proceed through these four steps: 1) change Set Up parameters from default values as needed; 2) change Display

Options parameters from default values as needed; 3) run the scenario with the Start selection on the Execution submenu; and 4) store any file output data needed from the run in a safe place (by renaming the file or changing the default file output name) so later runs do not overwrite it. When a run is complete, parameters can again be changed as needed, and a new scenario run, or the program can be exited.

Set Up Submenu Selections. The Set Up submenu controls the orbital scenario to be analyzed by the program.

Constellation. Specifies the drive, directory, and filename containing NORAD two-line element set data for the satellites to be used in the run. If the drive specification is left blank, the program will look in the current drive.

Observer Frame. Specifies the reference frame of the observer, either earth-centered inertial (ECI), rotating with the earth, or on one of the satellites.

Observer Location. Specifies the latitude (-90 to 90 degrees), longitude (0-360 degrees east longitude), and height above the earth's surface (in kilometers) of the observer at the start of the run. In the inertial frame (ECI), this will change as the earth rotates. It will remain fixed in the rotating earth frame. This input is ignored when the observer is on a satellite, the program instead uses the satellite location as the observer location.

Set Time. Specifies the start and stop dates and times for the scenario and the time increment between each calculation of satellite position. Dates are in the format 01JAN1991 and times are in the format 1300 (HHMM). The time increment is in minutes (decimal fractions of a minute are also accepted).

File Output. Allows an output file name and format to be specified. During execution data will be output to this file for later analysis by hand or some other computer program. The output file name is specified the same as in the Constellation selection.

Three output formats are available. Inputting 0 gives no file output. Format 1 outputs current simulation time (in format 01 JAN 1991 at 130000, i.e., DD MON YYYY at HHMMSS) plus the earth-centered inertial (ECI) coordinates of the satellite at that time (rectangular coordinates, in kilometers). Format 2 outputs current time plus the latitude, longitude, and height above the surface of the earth of the satellite (in degrees and kilometers). Format 3 outputs current time plus look angles (azimuth, elevation, slant range) from a selected earth point to the satellite. Azimuth is measured from north, clockwise, in degrees. Elevation is measured from the horizon, toward the zenith, in degrees. Negative elevations in the output indicate the satellite was below the horizon, and was not visible. The slant range is the distance from the earth point on the earth's surface to the satellite, in kilometers.

The program also requires a satellite number to be input, an integer indicating which satellite in the satellite file is to have data for it saved. An input of 1 indicates the first satellite in the file, 2 indicates the second satellite in the file, and so on. An earth point number must also be specified. This is the number of the earth point (a point on the surface of the earth specified in the Point On Earth selection of the Display Options menu) which will be used in calculations for Format 3. It is ignored for the other formats. Only one satellite and one earth point may be specified at a time. To gather more data, run the program again with different satellites or points. Change the output filename for each run, or old output will be erased and overwritten by subsequent runs.

A final input on the File Output screen is for a descriptor. This is a user defined phrase which will be placed at the top of the output file. It should be used to label the output so it can be identified at a later date. One possible format appears as a default, giving constellation and satellite number. This default is calculated from the constellation and satellite picked automatically, so changes in a descriptor for one run will not be stored as a new default for the next run.

In all cases, file output is stored in the output file in ASCII form, one line per time increment with no labels except the descriptor at the top and the time string label at the start of each line. The data is separated from the time string and each other by two spaces each. A typical Format 1 file would look like this:

```
GPS2.TLE Sat 1 Output        <-  descriptor
01 JAN 1991 at 010000   23426.867   -4194.055   11487.063
01 JAN 1991 at 011500   24764.267   -2175.376    8957.323
01 JAN 1991 at 013000   25668.934    -118.686    6271.034
01 JAN 1991 at 014500   26125.047    1940.079    3475.140
       ^                    ^           ^           ^
       |                    |           |           |
  <time string>       <x coord.>   <y coord.>  <z coord.>
```

Reset Defaults. This selection will reset the DEFAULT.DTA default file to values hardcoded into the program. This is often quicker than undoing multiple changes made for the last run, and provides a reliable starting point from which to make changes. When selected, the drive will come on to write the new defaults to the file. No data input screen will be shown.

Return To Main Menu. This selection exits the submenu and returns to the main menu for further selections. All exit selections are at the bottom of their menus so they can be easily reached by holding down the down arrow key (the selection bar will not scroll past the exit to the top again).

Display Options Submenu Selections. The Display Options submenu controls how the program output will appear on the screen.

Grid. Specifies whether a latitude, longitude grid will be turned on and displayed. ON displays the grid, OFF does not. Grid lines are displayed every 20 degrees of latitude and every 30 degrees of longitude.

Map. Specifies whether a map will be turned on and displayed. ON displays the map, OFF does not. Displaying the map slows the program down substantially. If only one point or area is of interest, use the Point on the Earth selection instead to identify the point of interest exactly, or place earth points around the border of the area of interest. Running the scenario at the time of interest wastes less time than starting it hours early and having to wait for the desired time to arrive.

Change Magnification. Specifies the magnification to be used in displaying the scenario without affecting the observer height which determines how much of the earth will be visible. A magnification of 1.0 should give a picture on the screen about the same angular size as the observer would really see. This 'real size' assumes an eye to screen distance of 18 inches, on a 12" diagonal monitor. Magnifications greater than 1.0 give a larger image, smaller than 1.0 give a smaller image. Adjusting the magnification is often critical to get good displays when low flying satellites are involved. Use a less than 1.0 magnification when the observer is on the low orbit satellite to see a larger field of view. Use a distant observer with a large magnification in an ECI or rotating frame so low flying satellites are not blocked by the earth's horizon.

Orbit Parameters. Specifies which satellite number orbit parameters will be displayed on screen during execution. Input is 0 for none, 1 for the first satellite in the file, 2 for the second, and so on. Parameters displayed will be epoch, semi-major axis (a, in km), eccentricity (e), inclination (i, in degrees), right ascension of the ascending node (RA, in degrees), argument of perigee (w, in degrees), and mean anomaly (M, in degrees). Either orbit parameters or satellite location may be displayed, but not both.

Satellite Location. Specifies for which satellite number the satellite's location will be displayed on screen during execution. Input is 0 for none, 1 for the first satellite in the file, 2 for the second, and so on. Parameters displayed will be satellite number, satellite ECI position (in km) and velocity (in km/sec). Either orbit parameters or satellite location may be displayed, but not both.

Ground Track. Specifies for which satellite number a ground track will be displayed during execution. Input is 0 for no ground track, 1 for the first satellite in the file, and so on.

Point On Earth. Specifies 0 - 15 locations on the earth to be displayed during execution. The first input screen asks for how many locations (0 for none, up to 15) and subsequent screens ask for the latitude (-90 to 90 degrees) and longitude (0-360 degrees, east longitude) of each earth point. They will be displayed in magenta along with a point number while visible to the observer.

Return To Main Menu. This selection exits the submenu and returns to the main menu for further selections.

Execution Submenu Selections. The Execution submenu controls execution and limited real-time modifications to the scenario.

Start. This selection starts execution of the scenario, from the start time if a new scenario, or from the current time if the scenario was paused in progress. A scenario in progress can be modified with Pan or Zoom, or paused by hitting any key, and restarted with the Start selection. Other modifications require the scenario to be stopped, modifications made from the main menu, and restarted from the beginning. However, this is a quick process since the program remembers the last values of input parameters and uses them as new defaults. When the scenario stop time is reached, execution automatically stops. Hit any key to return to the main menu.

Pan. This selection allows the location of the observer to be changed during execution by specifying new latitude and longitude coordinates for the observer. The defaults shown will be current values. This option cannot be used if the observer reference frame is on a satellite.

Zoom. This selection allows the location of the observer to be changed during execution by specifying a new height for the observer. The default shown will be the current value. This option cannot be used if the observer reference frame is on a satellite. Zoom changes the location of the observer from which projections are done, while Change Magnification does not move the observer, it only simulates looking through a telescope of some power from the same location.

Stop. This selection stops program execution, after which it cannot be restarted from the current time. Once execution has stopped, hit any key to return to the main menu.

Return To Main Menu. This selection exits the submenu and returns to the main menu for further selections. All exit selections are at the bottom of their menus so they can be easily reached by holding down the down arrow key (the selection bar will not scroll past the exit to the top again). A simulation run in progress must be stopped before Return to Main Menu is selected. This selection is primarily used to exit from the submenu before execution is started, since the program will return to the main menu directly when it is stopped or when the stop time is reached.

Display Symbols. The display shows the satellites as they move in their orbits around the earth. In general, current satellite locations are in yellow, past locations are in brown. However, the current location of the satellite generating the ground track is green vice yellow. Only the last 50 locations for each satellite are stored; if more of the orbit is desired, increase the time increment. The satellite's ground track shows up in green, and earth points selected in Point On Earth show up in magenta (purple) with a number label. The grid displays circles of latitude every 20 degrees and circles of longitude every 30 degrees. Only that portion of the earth visible to the observer is displayed; the horizon shows up as a white circle in the center of the display. A light gray (almost white) + in the screen's center shows the center of the earth. If the display is not as desired, adjust the various control parameters (frame, observer location, magnification) until the screen shows the desired information.

Troubleshooting. Some problems are easy to fix. Some possible ones follow.

If the program will not run, check that the current directory and the program's directory are both \TP\SATMAP3\. Check that all files are there, including SATMAP3.EXE, DEFAULT.DTA, GPS2.TLE or whichever other satellite constellation file is currently in the default file, WMAP-NEW.DTA, and EGAVGA.BGI. The default file DEFAULT.DTA is easily destroyed if incorrect parameters are entered during execution. If it is not there, or empty, create a new one from a copy kept on the disk, using the command:

COPY DEFACOPY.DTA DEFAULT.DTA

The satellite file (one ending in .TLE) must not have more than 50 satellites in it. Sometimes merely running the program again will work if not all of the files were opened or closed properly.

If the program will not accept an input parameter, double check the format described on the screen below the input lines.

If the output file is empty when it should contain information, remember to rename it or pick a new output file name after each run, or before resetting defaults (resetting defaults is only a problem if the name OUTPUT.OUT was used).

The Default File. The default file DEFAULT.DTA is updated after each selection so that new settings become the program defaults. This file can be reset to values hard coded in the program with the Reset Defaults option. To allow the possibility of external editing, the file's contents and their purpose are reprinted below.

```
                                satellite file drive (blank)
\TP\SATMAP3\                    satellite file directory
GPS2.TLE                        satellite file filename
-1                              observer frame (ECI)
 0.00000000000000E+0000         observer latitude (radians)
 0.00000000000000E+0000         observer longitude (radians)
 2.00000000000000E+0005         observer height (km)
01/0000 JAN 1991                start date
02/0000 JAN 1991                stop date
 1.50000000000000E+0001         time increment (min)
                                output file drive (blank)
\TP\SATMAP3\                    output file directory
OUTPUT.OUT                      output file filename
0                               output file format (none)
1                               output satellite #
1                               output earthpoint #
0                               grid (0=no grid,1=yes grid)
0                               map  (0=no map ,1=yes map)
 1.00000000000000E+0000         magnification
0                               orbit parameter satellite #
0                               satellite location sat #
1                               ground track satellite #
5                               number of earth points
 0.00000000000000E+0000         latitude (rads)  of earth point 1
 0.00000000000000E+0000         longitude (rads) of earth point 1
 0.00000000000000E+0000         latitude (rads)  of earth point 2
 1.57079632679490E+0000         longitude (rads) of earth point 2
 0.00000000000000E+0000         latitude (rads)  of earth point 3
 3.14159265358979E+0000         longitude (rads) of earth point 3
 0.00000000000000E+0000         latitude (rads)  of earth point 4
 4.71238898038469E+0000         longitude (rads) of earth point 4
 5.23598775598299E-0001         latitude (rads)  of earth point 5
 0.00000000000000E+0000         longitude (rads) of earth point 5
 5.23598775598299E-0001         latitude (rads)  of earth point 6
 1.57079632679490E+0000         longitude (rads) of earth point 6
 5.23598775598299E-0001         latitude (rads)  of earth point 7
 3.14159265358979E+0000         longitude (rads) of earth point 7
 5.23598775598299E-0001         latitude (rads)  of earth point 8
 4.71238898038469E+0000         longitude (rads) of earth point 8
 9.59931088596881E-0001         latitude (rads)  of earth point 9
 0.00000000000000E+0000         longitude (rads) of earth point 9
 9.59931088596881E-0001         latitude (rads)  of earth point 10
 1.57079632679490E+0000         longitude (rads) of earth point 10
 9.59931088596881E-0001         latitude (rads)  of earth point 11
 3.14159265358979E+0000         longitude (rads) of earth point 11
 9.59931088596881E-0001         latitude (rads)  of earth point 12
 4.71238898038469E+0000         longitude (rads) of earth point 12
-1.04719755119660E+0000         latitude (rads)  of earth point 13
 0.00000000000000E+0000         longitude (rads) of earth point 13
-1.04719755119660E+0000         latitude (rads)  of earth point 14
 1.57079632679490E+0000         longitude (rads) of earth point 14
-1.04719755119660E+0000         latitude (rads)  of earth point 15
 3.14159265358979E+0000         longitude (rads) of earth point 15
```

Satellite .TLE Files. NORAD two-line element sets describing the orbits of the satellites to be modeled in the scenario are stored in files with the suffix .TLE. A single .TLE file can contain element sets for up to 50 satellites. Three lines of data are used for each satellite. The first contains the satellite's name, which is ignored by the program. The second and third lines are the standard NORAD two-line element sets with a format and interpretation described below (5).

Element Set Format

```
1 NNNNNU NNNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N+NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNNNN.NNNNNNNNNNNNNN
```

Line 1 Interpretation

Columns	Description
01 - 01	Line Number of Element Data
03 - 07	Satellite Number
10 - 11	International Designator (Last two digits of launch year)
12 - 14	International Designator (Launch number of the year)
15 - 17	International Designator (Piece of launch)
19 - 20	Epoch Year (Last two digits of year)
21 - 32	Epoch (Julian Day and fractional portion of the day)
34 - 43	First Time Derivative of the Mean Motion or Ballistic Coefficient (Depending on ephemeris type)
45 - 52	Second Time Derivative of Mean Motion (decimal point assumed; blank if N/A)
54 - 61	BSTAR drag term if GP4 general perturbation theory was used. Otherwise, radiation pressure coefficient. (Decimal point assumed)
63 - 63	Ephemeris type
65 - 68	Element number
69 - 69	Check Sum (Modulo 10) (Letters,blanks,periods = 0; minus sign = 1)

Line 2 Interpretation

Columns	Description
01 - 01	Line Number of Element Data
03 - 07	Satellite Number
09 - 16	Inclination, i (Degrees)
18 25	Right Ascension of the Ascending Node,RA (Degrees)
27 - 33	Eccentricity, e (decimal point assumed)
35 - 42	Argument of Perigee, w (Degrees)
44 - 51	Mean Anomaly, M (Degrees)
53 - 63	Mean Motion, n (Revs per day)
64 - 68	Revolution number at epoch (Revs)
69 - 69	Check Sum (Modulo 10)

Note that the semi-major axis, a, can be calculated from the mean motion, n, and vice versa, from a = (mu/(n^2))^(1/3) where mu = 3.986005x10^14 m^3/s^2 , n must be converted to radians per sec, and a converted to meters.

These satellite .TLE files can be easily browsed or edited using MS-DOS EDLIN commands or a text editor capable of importing/exporting files in ASCII format. The satellite .TLE files can also be updated with a specially designed program called UPDATE written by Dr. T.S. Kelso, and available from:

AFIT/ENS
Wright-Patterson AFB, OH, 45433-6583
Attn: GSO Program Director
DSN 785-3362

Satellite element set files for many satellites, updated weekly, are available through E-mail (electronic mail) via anonymous ftp (file transfer protocol) from blackbird.afit.af.mil in the directory pub/space. They are also uploaded to sci.space and rec.radio.amateur.misc.

SATMAP3 Quick Reference Section.

Menu Summary.

Set Up Submenu

Constellation	- Filename of satellite constellation
Observer Frame	- ECI, rotating with earth, or on sat.
Observer Location	- Latitude, longitude, height
Set Time	- Start, stop time; time increment
File Output	- Output data to file in 3 formats
Reset Defaults	- Set defaults to hardcoded values
Return To Main Menu	- Go back to main menu

Display Options Submenu

Grid	- Display lat./long. grid
Map	- Display earth map
Change Magnification	- Observer 'view' magnification
Orbit Parameters	- Display orbit parameters real-time
Satellite Location	- Display sat. x,y,z real-time
Ground Track	- Display sat. ground track
Point on Earth	- Define 0-15 points for display
Return To Main Menu	- Go back to main menu

Execution Submenu

Start	- Start scenario execution
Pan	- Modify observer lat./long.
Zoom	- Modify observer height
Stop	- End scenario early
Return To Main Menu	- Go back to main menu

(hit any key to pause, use start to restart)

Exit Program Selection

Active Keys.

Up and down arrows	- scroll through menu
ENTER	- selects a menu item or enters a data item
Right and left arrows	- position cursor within a data field for editing
Backspace	- same as left arrow
A-Z, a-z	- input text in uppercase
0-9, + , - , .	- input number or symbol
\ , : , .	- input symbol used in file names
Spacebar	- blanks out a character
<ESC> or Control-X	- returns field to value before editing

Appendix B: Suggestions on Use

This appendix provides suggestions on how to use SATMAP3 as an instructional and analytic tool. The section on instructional use includes a description of satellite element set files included with the program and designed to demonstrate various orbital parameters, orbit types, and typical space systems. It details the parameters which should be used in each case to obtain a clear display of the concept in question. The section on use in analysis presents a few ideas conceived by the author as to how the various features of the program can be used creatively to solve analytic problems.

Use in Instruction.

SATMAP3 should be used in teaching orbital mechanics and satellite operations wherever access to an IBM-compatible personal computer makes its use possible. In teaching orbital mechanics, its use will reveal the dynamic nature of satellite orbits as no paper drawing can. When used to teach satellite operations, SATMAP3 will add a memorable perspective on the nature of various orbits. The program disk contains a database of satellite element set files designed to be used to demonstrate various orbit parameters, types of orbits, and typical space systems. The following sections describe how to use this database to best advantage.

<u>Orbital Parameters</u>. Six satellite files were developed to demonstrate the orbital elements semi-major axis, eccentricity, inclination, right ascension of the ascending node, argument of perigee, and mean anomaly. Once the program has been installed and started as described in the user's guide, make menu selections as described below to obtain a clear display of each parameter.

<u>Semi-major Axis</u>. First orbit (middle one) is semi-synchronous (12-hour period), smallest orbit is a low earth orbit (100-minute period), largest is geosynchronous orbit (24-hour period).

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	SEMIMA.TLE
Set Up	Set Time	Increment	30
Display Opt.	Change Mag.	Magnification	0.8

<u>Eccentricity</u>. Three orbits with eccentricities of 0.0, 0.3, and 0.7.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	ECCENT.TLE
Display Opt.	Change Mag.	Magnification	0.8

<u>Inclination</u>. Three orbits with inclinations of 0, 30, and 60 degrees.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	INCLIN.TLE

Right Ascension of the Ascending Node. Three orbits with right ascensions of 0 (left most ascending node on the display), 30, and 90 degrees.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	RIGHTA.TLE
Set Up	Observer Loc.	Longitude	260

Argument of Perigee. Three orbits with arguments of perigee of 0 (semi-major axis horizontal on the display), 30, and 90 (semi-major axis vertical on the display) degrees.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	ARGUME.TLE
Display Opt.	Change Mag.	Magnification	0.8

Mean Anomaly. Three orbits differing only by the mean anomaly of the satellites in them, in this case mean anomalies of 0, 30, and 90 (leading satellite) degrees.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	MEANAN.TLE
Display Opt.	Change Mag.	Magnification	0.8

Orbit Types. Four satellite files were developed to demonstrate various special orbits. These include the following: 1) a sun-synchronous, polar, low-earth orbit typical of many weather satellites (in this case a NOAA satellite); 2) a Molniya orbit; 3) a semi-synchronous orbit (in this case a GPS navigation satellite with an inclination of 54 degrees); and 4) two geosynchronous orbits with inclinations of 1 and 10 degrees.

<u>Sun Synchronous Orbit</u>.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	SUNSY.TLE
Set Up	Set Time	Increment	5

<u>Molniya Orbit</u>.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	MOLNIYA.TLE
Set Up	Set Time	Stop Date	03JAN1991
Display Opt.	Change Mag.	Magnification	0.8

<u>Semi-Synchronous Orbit</u>.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	SEMISY.TLE

<u>Geosynchronous Orbit</u>. Use these changes to show the orbits.

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	GEOSY.TLE
Set Up	Set Time	Increment	30

<u>Geosynchronous Orbit</u>. Use these changes to show the ground track of an orbit with a positive inclination (10 degrees).

Menu	Submenu	Parameter	Value
Set Up	Reset Defaults		
Set Up	Constellation	Filename	GEOSY.TLE
Set Up	Observer Frame	Obs. Frame	0
Set Up	Observer Location	Longitude	225
Display Opt.	Change Mag.	Magnification	8.0
Display Opt.	Ground Track	Satellite #	2

Space Systems. The database included with the program also includes satellite element set files for a sample of current space systems. In each case, reset the defaults using the Set Up submenu and the Reset Defaults selection. Then change the constellation to the desired one using the Set Up menu, Constellation selection, at the filename parameter. A list of the included files, by filename, with a description of their contents now follows. Note that these constellations are not necessarily complete or up to date, especially their epoch dates. For analysis, element sets accurate at the time in question must be obtained.

Filename	Description
NOAA.TLE	3 NOAA civilian weather satellites, low orbit
DMSP.TLE	DoD weather constellation, low orbit, 3 sats
STS.TLE	4 typical space shuttle orbits
GPS2.TLE	GPS DoD navigation constellation, the first 10 Block II satellites only
GPS.TLE	GPS DoD navigation constellation, Block I and II satellites, not all working
GOES.TLE	2 GOES civilian weather sats, geosynchronous
DSCS.TLE	NATO and DSCS III [illegible] communications sats

Use in Analysis.

The following five comments or suggestions may be of aid in using SATMAP3 for analysis of orbital problems: 1) tailor the constellation of satellites in the satellite file to the problem at hand; 2) use an appropriate observer reference frame; 3) choose a time increment carefully; 4) use points on the earth for accuracy; and 5) generate output files to be the input for needed calculations.

Tailor the Constellation. The satellite file specified in the Constellation selection should include exactly those satellites needed in the study. Edit the file using a text editor with an ASCII export capability or the MS-DOS EDLIN editor to include all the satellites needed, but no more, to reduce clutter. Develop multiple files, each with different parameters, to compare different alternatives. The satellite file can contain a maximum of 50 satellites. If a study required more than this, the program could be modified and recompiled. Increase the constant MFSize to the new number of satellites and reduce the constant MNPoints (the number of old satellite locations stored for each satellite) until the program runs without running out of memory to store the variable array old_sat_pos.

Select the Observer Reference Frame. While the earth-centered inertial (ECI) frame is best for a quick look at a situation, there are times when the other frames should be used. Use a frame rotating with the earth to see what occurs in the vicinity of some location on the earth. Use an observer located on a satellite to show coverage. Only that part of the earth visible to the satellite will be displayed. When the ground track is turned on for that satellite, it is easy to see the relationship between the satellite and a point on the earth. It would be possible, for example, to see when a ground point falls within some swath of a satellite's ground coverage.

Choose the Time Increment. A carefully chosen time increment will avoid wasted time and provide the needed accuracy. Use long time increments (15, 30, 60 minutes or more) to get a quick look at the situation. Then reduce the time increment while adjusting the start time to more closely bracket the time of interest. Time increments of 1 minute or less provide more accuracy when it is needed in finding a rise time or time of closest approach. Such short time increments are also useful in comparing results to data taken at some specific time. A minute difference in the data can mean 10s or 100s of kilometers difference in location.

Use Earth Points For Accuracy. Definition of points on the earth to be displayed during execution, done using the Points on the Earth selection of the Display Options submenu, allows more accuracy than using the grid or map. A point on the earth is not displayed unless it is visible to the observer. As a point on the earth comes into view from below the horizon, the point itself may be hard to make out on the display since it is so close to the bright earth horizon circle, but the point label is easy to see. Placing the observer on the satellite, watching for the first appearance of the point label, and noting the time gives the rise time of the satellite with respect to that earth point without having to use the file output feature. Use the earth points to locate ground stations or targets of interest, or to bound an area of interest with a number of points.

Generate Output Files. Use the File Output option on the Set Up submenu to create output files for use in later calculations. Each scenario run can only generate one output file, concerning only one satellite, using only one type of coordinates, but the scenario can be run over again to generate different files for different satellites. User-written programs to process this data can then read in the data from more than one output file to perform the needed comparisons. Since these output files are stored in ASCII format they can be accessed by user written programs using almost any programming language. They could also be read into a spreadsheet program for analysis. The user is accessing the output files, not making modifications to SATMAP3, so he need only be concerned about the format of the output file, not with how his modifications might affect the rest of SATMAP3.

One simple post processor might be a program to calculate the distance between two satellites and the time of their closest approach. It would need to read in data from the two output files and find the satellite to satellite vector by subtracting the ECI coordinates of one satellite from the other. The magnitude of this vector would be the distance between the two satellites. It could be output with a time tag, or stored and compared to other distances to find the time and distance of closest approach.

Another simple post processor might use look angle output files (Format 3) to calculate visibility times. It could further process the data to include only those times when the elevation was above some minimum elevation important to that satellite program. Similarly, a program could use Format 2 output files (latitude, longitude, height) to calculate coverage of some point on the earth by satellites with some particular ground swath.

As can be seen from the above examples, the various features of SATMAP3 can be combined to conduct a wide variety of analyses. The program is flexible enough to work with many different types of satellites. In almost all cases, the needed scenario can be run and examined without the need to modify and compile SATMAP3, since input satellite files can be generated using most text editors, and post processors can be written in almost any language to access the ASCII output files the program generates.

Bibliography

1. Bruce, L.B. "Screen to Printer Dumps in Pascal for the IBM-PC 320x200 Color Graph Mode," Journal of Pascal, Ada & Modula-2, 7: 5-12 (July/August 1988).

2. Eagle, David. "Computing Crossing Data for Earth Satellites," Electronics, 56: 122-123 (January 1983).

3. Gupta, Satish and Daniel H. McCabe. "Personal Computer Displays," IEEE Computer Graphics & Applications, 7: 17-23 (October 1987).

4. Hoots, Felix R. and Ronald L. Roehrich. Spacetrack Report No. 3, Models for Propagation of NORAD Element Sets, package compiled by T.S. Kelso. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988.

5. Kelso, Thomas S. Class handout distributed in OPER 592, Space Operations Planning. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, February 1991.

6. Kelso, Thomas S., Director, Graduate Space Operations Program. Personal interviews. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1 April through 22 April 1991.

7. Kelso, Thomas S. SAT-MAP2. Computer program which plots a view of the earth and an orbiting satellite. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, November 1989.

8. Lawton, J.A. "Numerical Method for Rapidly Determining Satellite-Satellite and Satellite-Ground Station In-View Periods," Journal of Guidance, Control, and Dynamics, 10: 32-36 (January-February 1987).

9. Ledgard, Henry and John Tauer. Professional Software, Volume 1. Reading, Mass.: Addison-Wesley Publishing Company, 1987.

10. PC SOAP Display Module. Computer program for displaying Global Positioning System (GPS) satellite orbits. The Aerospace Corporation. 1988.

11. Plastock, Roy A. and Gordon Kalley. Schaum's Outline of Theory and Problems of Computer Graphics. New York: McGraw-Hill Publishing Company, 1986.

12. Putney, B. and others. "Precision Orbit Determination at the NASA Goddard Space Flight Center," Advances in Space Research, Vol. 10, No. 3-4: 197-203 (1990).

13. Raol, J. R. and N. K. Sinha. "On the Orbit Determination Problem," IEEE Transactions on Aerospace and Electronic Systems, AES-21: 274-291 (May 1985).

14. Tang, Charles C. H. "The Exact Solution for Orbit View-Periods From a Station on a Tri-Axial Ellipsoidal Planet," The Journal of the Astronautical Sciences, 35: 447-460 (October-December 1987).

15. Yip, B. and others. "Three-Dimensional Gas Concentration and Gradient Measurements in a Photoacoustically Perturbed Jet," Applied Optics, 25: 3919-3923 (November 1986).

16. Yorchak, J. P. and others. "An Experimental Plan for Assessing the Advantages of Using Interactive Computer Graphics in Training Individuals in Basic Orbital Mechanics," Human-Computer Interaction, edited by G. Salvendy. Amsterdam: Elsevier Science Publishers B. V., 1984.

17. Ziegler, J. "Information Design for Screens and Screen Menus," Software Ergonomics Advances and Applications, edited by H.-J. Bullinger and R. Gunzenhauser. New York: Halsted Press, 1988.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Master's Thesis

4. TITLE AND SUBTITLE
A DYNAMIC COMPUTER GRAPHICS MODEL OF SATELLITE ORBITS FOR USE IN INSTRUCTION AND ANALYSIS

5. FUNDING NUMBERS

6. AUTHOR(S)
John P. Anton, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSO/ENS/91D-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)
A computer model of satellite orbits was developed to illustrate difficult orbital concepts. It was also designed to analyze the positions of a satellite constellation with respect to locations on the earth as an aid to analysts. A literature review revealed that existing methods rely on drawings, expensive computers, or are satellite system specific, but that it would be possible to develop such a tool for an IBM-compatible personal computer using the Pascal language. Model capabilities selected included 1) a menu driven user interface; 2) a screen display showing satellite locations, ground track, and an earth map in one of three reference frames; and 3) three file output formats allowing satellite position information to be dumped to ASCII files for further analysis. Model validation was performed to ensure that the satellites are accurately propagated in their orbits from the NORAD two-line element sets used as an input to the model. The model should be used in instruction to bring the dynamic nature of satellite orbits to life and add a memorable perspective on the nature of various constellations. In analysis, it should be used where convenience, availability, and ease of use are vital.

14. SUBJECT TERMS
Computer Graphics, Computerized Simulation, Artificial Satellites, Man Computer Interfaces, Orbits, Map Projection, Mapping (Transformations)

15. NUMBER OF PAGES
85

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to **stay within the lines to meet optical scanning requirements.**

Block 1. Agency Use Only (Leave Blank)

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract
G - Grant
PE - Program Element
PR - Project
TA - Task
WU - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Names(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency. Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of ..., To be published in .... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availablity Statement. Denote public availability or limitation. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR)

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - DOD - Leave blank
DOE - DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports
NASA - NASA - Leave blank
NTIS - NTIS - Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.